What is claimed is:

1. An application-independent system for dynamically generating low-complexity graphics embedded as Web content using a tag-delimited script, comprising:

a stored object library specifying a graphics object class defining a logical canvas and comprising a vector of row objects which each contain a set of cell objects each defining display attributes for a uniform rectangular region;

a set of methods, each encapsulated in the graphics object class and drawing a shape onto the logical canvas by sequentially parsing through each row object in the vector and through each cell object contained therein to consistently structure the logical canvas; and

a rendering engine that converts the logical canvas into a table encoded in a tag-delimited script by converting in order each row object into a row within the table, and each cell object into a cell within each row.

2. A system according to Claim 1, further comprising:

a routine resolving overlap between at least one of a plurality of adjacent row objects and a plurality of adjacent cell objects.

3. A system according to Claim 2, wherein the routine splits overlapping cell objects into a plurality of non-overlapping cell objects.

4. A system according to Claim 2, wherein the routine splits overlapping row objects into a plurality of non-overlapping row objects.

5. A system according to Claim 1, further comprising:

a shape builder selecting the display attributes from a list comprising at least one of color, text, text style, dimensions, location, hyperlink, and bubble text.

6. A system according to Claim 1, further comprising:

a shape builder creating shapes comprising at least one of a clear rectangle, three-dimensional rectangle, arc, characters, image, line, oval, polygon,

polyline, rectangle, round rectangle, string, filled three-dimensional rectangle, filled arc, filled oval, filled polygon, and filled round rectangle.

7. A system according to Claim 1, further comprising:

an application programming interface used by a Java Abstract Windows-compliant toolkit exported by the stored object library.

8. A system according to Claim 1, further comprising:

a Web server sending rendered graphics as a downloaded HTML script.

9. An application-independent process for dynamically generating low-complexity graphics embedded as Web content using a tag-delimited script, comprising:

specifying a graphics object class defining a logical canvas and comprising a vector of row objects which each contain a set of cell objects defining display attributes for a uniform rectangular region;

drawing a shape onto the logical canvas by sequentially parsing through each row object in the vector and through each cell object contained therein to consistently structure the logical canvas; and

converting the logical canvas into a table encoded in a tag-delimited script by converting in order each row object into a row within the table, and each cell object into a cell within each row.

10. A process according to Claim 9, further comprising:

resolving overlap between at least one of a plurality of adjacent row objects and a plurality of adjacent cell objects.

11. A process according to Claim 10, further comprising:

splitting overlapping cell objects into a plurality of non-overlapping cell objects.

12. A process according to Claim 10, further comprising:

splitting overlapping row objects into a plurality of non-overlapping row objects.

13. A process according to Claim 9, further comprising:

selecting the display attributes from a list comprising at least one of color, text, text style, dimensions, location, hyperlink, and bubble text.

14. A process according to Claim 9, further comprising:

creating shapes comprising at least one of a clear rectangle, three-dimensional rectangle, arc, characters, image, line, oval, polygon, polyline, rectangle, round rectangle, string, filled three-dimensional rectangle, filled arc, filled oval, filled polygon, and filled round rectangle.

15. A process according to Claim 9, further comprising:

exporting an application programming interface used by a Java Abstract Windows-compliant toolkit.

16. A process according to Claim 9, further comprising:

sending rendered graphics as a downloaded HTML script.

17. A computer-readable storage medium holding code for performing the process according to Claims 9, 10, 11, 12, 13, 14, 15, or 16.

18. An application-independent system for specifying graphical Web content as a table written in a tag-delimited script, comprising:

an object library storing a graphics object class including state variables and at least one shape method, the graphics object class comprising:

a vector as one such state variable comprising at least one row object reference, each comprising at least one cell object reference; and

encoded logic encapsulated within the graphics object class defining the at least one shape method to generate a selectable shape within a logical canvas; and

a rendering engine rendering a table as a tag-delimited script from an instance of the graphics object class containing the selectable shape rendered from each at least one row object reference and each at least one cell object reference.

19. A system according to Claim 18, further comprising:

an application programming interface exported from the graphics object class referencing the at least one shape method.

20. A system according to Claim 18, further comprising:

a constructor method instantiating an instance of the graphics object class.

21. A system according to Claim 18, further comprising:

a shape builder storing a uniform rectangular area into each such at least one cell object reference; and

a renderer generating each uniform rectangular area as a cell in a row in the table.

22. A system according to Claim 18, further comprising:

a shape builder specifying location, dimensions and related characteristics and attributes for the selectable shape.

23. A system according to Claim 18, further comprising:

a shape builder specifying at least one such shape method to generate one such selectable shape selected from at least one of a clear rectangle, three-dimensional rectangle, arc, characters, image, line, oval, polygon, polyline, rectangle, round rectangle, string, filled three-dimensional rectangle, filled arc, filled oval, filled polygon, and filled round rectangle.

24. A system according to Claim 18, further comprising:

a routine determining overlap a plurality of rendered selectable shapes within the logical canvas.

25. A system according to Claim 24, wherein the routine splits horizontally overlapping regions into a plurality of non-overlapping regions within one such cell object reference.

26. A system according to Claim 24, wherein the routine splits vertically overlapping regions into a plurality of non-overlapping regions within one such row object reference.

27. An application-independent process for specifying graphical Web content as a table written in a tag-delimited script, comprising:

defining a graphics object class including state variables and at least one shape method, comprising:

including a vector as one such state variable comprising at least one row object reference, each comprising at least one cell object reference; and

encoding logic encapsulated within the graphics object class defining the at least one shape method to generate a selectable shape within a logical canvas; and

rendering a table as a tag-delimited script from an instance of the graphics object class containing the selectable shape rendered from each at least one row object reference and each at least one cell object reference.

28. A process according to Claim 27, further comprising:

exporting an application programming interface from the graphics object class referencing the at least one shape method.

29. A process according to Claim 27, further comprising:

instantiating an instance of the graphics object class.

30. A process according to Claim 27, further comprising:

storing a uniform rectangular area into each such at least one cell object reference; and

generating each uniform rectangular area as a cell in a row in the table.

31. A process according to Claim 27, further comprising:

specifying location, dimensions and related characteristics and attributes for the selectable shape.

32. A process according to Claim 27, further comprising:

specifying at least one such shape method to generate one such selectable shape selected from at least one of a clear rectangle, three-dimensional rectangle, arc, characters, image, line, oval, polygon, polyline, rectangle, round rectangle, string, filled three-dimensional rectangle, filled arc, filled oval, filled polygon, and filled round rectangle.

33. A process according to Claim 27, further comprising:

determining overlap a plurality of rendered selectable shapes within the logical canvas.

34. A process according to Claim 33, further comprising:

splitting horizontally overlapping regions into a plurality of non-overlapping regions within one such cell object reference.

35. A process according to Claim 33, further comprising:

splitting vertically overlapping regions into a plurality of non-overlapping regions within one such row object reference.

36. A computer-readable storage medium holding code for performing the process according to Claims 27, 28, 29, 30, 31, 32, 33, 34, or 35.

ADD A8